SYSTEMS, SCIENCE AND SOFTWARE

3SR-405

ENERGY DEPOSITION IN LASERS

Bimonthly Technical Report
For Period 1 May 1970 - 30 June 1970

Principal Investigator: Dr. Marius Troost (714) 453-0060
Project Scientist: Mr. C. W. Hartley (ONR)

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 660

Program Code Number: 6 23 01 D
Contract No. N00014-70-C-0341
Effective Date of Contract: 1 May 1970
Contract Expiration Date: 31 December 1970
Amount of Contract: $44,000 S³ Project 130

29 July 1970

P O BOX 1620, LA JOLLA, CALIFORNIA 92037, TELEPHONE (714) 453-0060

TABLE OF CONTENTS

I. SUMMARY

A UNIVAC 1108 FORTRAN IV computer program has been written to determine the distribution of energy deposition within a LASER assembly. The code is currently running but needs further enhancement of its input and output facilities as well as a source description. The program uses ray tracing and Monte Carlo methods to obtain a solution. The geometrical and property input to the code is very simple and convenient to write, but the source input still needs to be defined.

II. INTRODUCTION

General Methods of Solution

Several methods might be employed to calculate the distribution of the energy deposition within a Laser. Analytical calculations are impossible due to the complexity of the geometry. Numerical solutions of the relevant wave equations should be possible, but the computer time cost for three-dimensional calculations is prohibitive. Ray tracing, which is frequently used in optical problems, seems to be the most practical method available.

Ray Tracing

In ray tracing, rays may be selected systematically or at random. In simple geometries, systematic selection of rays is advantageous as only few are needed for a good sample. In complicated, 3-dimensional geometries the number of rays required for an adequate systematic sampling becomes prohibitively large and randomly selected rays from known distributions will give more reliable

results for the same amount of computer time. As the proposed laser designs are of considerable complexity, the computer program will use randomly selected rays from specified distributions.

A problem for any method of ray tracing in a design with a large amount of internal reflection and no black absorbers, is to decide when to stop following a particular ray. As each ray loses its energy very gradually via partial absorption, it is impractical to follow the ray until it is completely absorbed; without a black absorber this would take a very large number of reflections. If a ray is ignored only after its energy falls below a negligible level (e.g. 1% of its original strengh) the paradoxical situation exists where most of the tracing effort is used for rays that contribute very little to the final result. If, on the other hand, each ray is cut-off when its energy is still significant (e.g. 25% of its original strength), large errors will be made. A method is needed that does not ignore weak rays, but that does not waste too much effort on them.

Monte Carlo and Russian Roulette

By randomly selecting the rays to be traced the problem is transformed into one that can be solved by Monte Carlo methods. A technique frequently used in Monte Carlo Solutions is the Russian Roulette Technique. This technique allows each ray to be cut off while its energy is still significant without ignoring the weak rays. Basically this method involves statistically eliminating some rays which are still quite energetic, while manufacturing other weaker rays for further tracing. In practice the Russian Roulette method works as follows.

A ray is followed until its intensity falls below a certain predetermined level (e.g. 50% of original intensity). At that point it is decided randomly whether this ray will survive. If it dies, it is ignored and no record is kept. If it survives, its intensity is multiplied by a factor which is the inverse of the probability of survival in the random test. The intensity of the surviving ray is therefore always relatively high (e.g. above 30% of the original intensity). As each ray may several times pass through this procedure before being killed, even very weak rays are statistically represented in the sample. Partial reflections may be treated by creating a new ray of appropriate intensity. A pass through the Russian Roulette routine will then decide whether the new ray survives and with what intensity.

Energy deposition is determined by calculating the exponential attenuation along the path of a ray. The deposition is assigned to the segment through which the ray passed. The intensity is checked at every intersection of the ray with a boundary, and a game of Russian Roulette is played when necessary. Eventually each ray will therefore be killed in a game of Russian Roulette.

The accuracy of the energy absorbed in the various regions of the system is dependent on the number of rays traced. Several runs may be necessary to determine the number of rays needed for the accuracy required.

III. GEOMETRY

Geometry Specifications

A flexible method for describing the geometry of the system has been developed which allows simple input and reduces

computer execution time. At the same time, it allows for greater generality in geometrical configuration.

A problem common to many Monte Carlo codes is their extreme slowness when applied to complicated geometries. This slowness is caused by the need to check for possible intersections with all possible boundaries. The running time thus becomes roughly proportional to the number of boundaries in the system. It becomes quickly evident that some technique is needed which eliminates the necessity for checking all boundaries for each intersection.

One possible solution to this problem is to divide space into subregions and to check only for possible intersections with those boundaries that exist in the subregion and with the boundaries of the subregion itself. Some modern Monte Carlo codes use this method but they are often inconvenient to use because of the cumbersome geometrical definitions required for each subregion.

S^3 has developed a method which uses the subdivision method recursively. The geometry is specified by a greatly simplified geometrical input, as the subregions used are the natural geometrical shapes of the system.

Segments

Each part of space in the problem to be studied is divided into one or more segments. A single piece of material may belong to several segments or a segment may contain several materials in the form of sub-segments. Two kinds of segments are defined: Mother Segments and Daughter Segments.

The Mother segments act as subregions and the Daughter segments lie within Mother segments. When a photon lies in a particular Mother segment, only the intersections with boundaries of that Mother segment and its Daughter segments need to be checked. The boundaries of the Mother segment shield the photon therefore from the details in the space outside the Mother segment.

It frequently is also desirable to shield the photon from details in certain areas within the Mother segment. This may be done by enclosing these details in a Daughter segment and declaring this Daughter segment to be a Mother segment at a lower level. We have now a recursive definition of Mother and Daughter segments, where any Mother segment may be a Daughter segment of a higher level Mother and any Daughter segment may be a lower level Mother segment.

The geometry of the complete system fits into one Mother segment. This Mother segment usually will have several Daughter segments that are Mother segments in turn.

This recursive subdivision may be continued to any desired depth. Any Daughter segment, that is not a Mother segment must consist of a homogeneous material.

A Mother segment is a segment which is partially or completely filled by one or more Daughter segments. That part of space within a Mother segment that is not occupied by Daughter segments has the material properties of the Mother segment. The volume of the Mother segment is the volume of the space within the Mother segment that is not occupied by Daughter segments.

All of the external boundaries of a Mother segment must be specified. Internal boundaries which define Daughter segments need not be specified in the discription of the Mother segment. This simplifies the problem input considerably.

Allowable Boundaries

The code is open ended with respect to boundary types. In principle any boundary that can be described mathematically may be added to the code by writing the appropriate geometrical subroutines and adding the new type to a table.

Currently the code contains routines that handle the following boundaries:

1. Plane (in any attitude)
2. Right circular cylinder (in any attitude)
3. Conic sections (cylindrical axis must be parallel to Z-axis)

Additional boundaries may be added to the code when required.

Representation of Geometry

The Mother-Daughter relationship is defined on segment input records. Each record contains the Mother segment number followed by the Daughter segment numbers in order of importance. Importance is defined such that a Daughter segment may be overlaid by any or all Daughter Segments to the right in the sequence but by none of the Daughter segments to the left in the sequence. The advantage of this hierarchy is that the number of boundaries that need be specified for each Daughter segment is reduced when

overlapping occurs. Also, when inside a Daughter segment only the boundaries of higher ranking Daughter segments need to be checked as the lower ranking Daughter cannot overlay this Daughter segment. Finally, after a boundary crossing, while determining which new segment has been entered, only the Daughter segments need to be checked if the photon is still within the Mother segment.

Boundary Intersections

A. When a photon is inside a Daughter Segment the following boundaries should be checked:

   1. The boundaries of the current Daughter segment.
   2. The boundaries of all higher ranking Daughter segments.
   3. The boundaries of the Mother segment.

B. When a photon is inside a Mother segment but not inside any of its Daughter segments, the following boundaries should be checked:

   1. The boundaries of all Daughter segments in this Mother segment.
   2. The boundaries of the Mother segment.

New Segment Determination

After a photon has crossed a boundary, the segment containing the photon must be determined. This search is conducted as follows:

   1. Is the photon still in the Mother segment? If the answer is yes,
   2. Check each Daughter segment, starting with the highest ranking one until one is found that contains the photon.

If the photon does not lie in any of the Daughter segments, it lies by definition in the Mother segment part of space. If the photon lies within a Daughter segment a check must be made to see if this Daughter segment is a Mother segment at a lower level. If such is the case, the procedure must be repeated starting at 2. for this new Mother segment.

If the photon lies outside the current Mother segment the procedure should be repeated starting with 1. for the next higher level Mother segment.

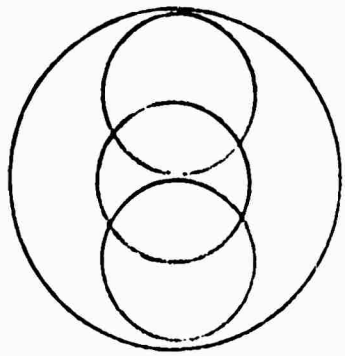
The search thus continues until the final segment containing the photon has been found. This might involve climbing several Mother levels and then descending several Mother levels. The Algorithm is written recursively and is not concerned with the absolute Mother level. When a photon is found outside the highest level Mother, an error stop will occur.

Selection of Mother Segments

Whenever possible, Mother segments should have simple shapes, e.g. rectangular parallelepiped, cylinder, sphere. A simple shape allows an efficient check for boundary crossings and reduces computer time. However, in principle a Mother segment may have any shape, even though that may be expensive in computer time.

Although the possible number of Daughter segments within a Mother segment is unlimited, in practice this number should be quite low, e.g. below 5, to reduce the number of boundary crossing checks. After all, the Mother segment was introduced to reduce the number of potential boundary crossings.

Through the use of the Segment Hierarchy scheme, it is possible to define several unique geometric systems from the same set of boundaries. Consider the four circular boundaries shown in Fig. 1-a. By changing the segment hierarchy we get the following results.

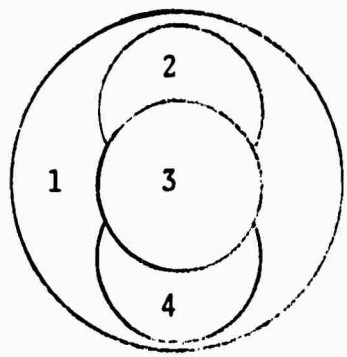


1-a

---

1) 1:2,4,3

This arrangement of Mother Segment #1, and Daughters 2,4,3 where each succeeding Daughter has precedence over those to the left, gives the system of segments shown in Fig. 1-b.

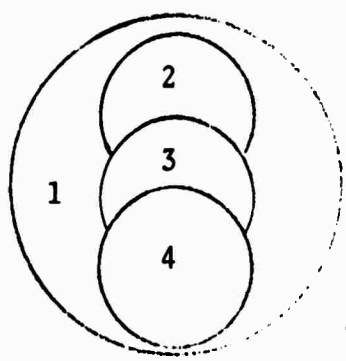


1-b

---

2) 1:2,3,4

This arrangement of Daughters will define the system shown in Fig. 1-c.



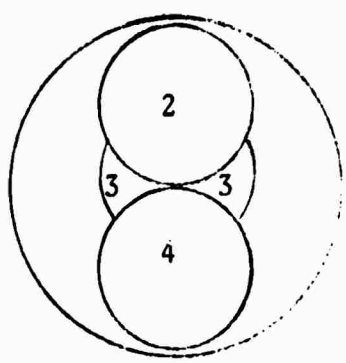
1-c

---

3) 1:3,2,4

Fig. 1-d shows the system generated by the above arrangement.

For each segment in the example only one circular boundary needs to be specified for that segment. Without the hierarchy scheme, all confining boundaries of each segment would have to be specified.



1-d

Figure 1

Examples of Segment Hierarchy

Daughter Segment Definition

A Daughter segment must lie completely within its Mother segment. Each Daughter segment is therefore associated with only one Mother segment. Each Daughter segment may in turn be a lower level mother segment. A Daughter segment may be partially overlaid by one or more higher ranking Daughters. The boundaries formed by such overlays need not in general be specified in the segment definition.

IV. GENERAL INPUT DESCRIPTION

The input to the code consists of:

1. General Information
2. Source Definitions
3. Mother-Daughter Hierarchy
4. Segment Definitions
5. Boundary Definitions
6. Material Definitions
7. Reflection Definitions

Each of these groups may consist of a variable number of input records. The sequence of the input records within a group is usually not important, but the sequence of the groups is fixed. Each group is separated from the next one by a blank input record. Each of these sections is described in detail below.

1. General Information

This information has not been specified in detail. Currently it consists of a Title and Options.

2. Source Definitions

The Source Definition input records have not yet been specified. They might consist of distributed volume sources or surface sources.

3. Mother-Daughter Hierarchy

Each input record contains:

a. A Mother Segment number.
b. Any number of Daughter Segment numbers.

a. Segment Number

Each segment number is defined by a positive integer greater than zero and less than the maximum number of segments allowed. (e.g. 1, 25, 42).

b. Daughter Segments

Each daughter segment may be partially or completely overlaid by any daughter segment specified later in the same geometry input record.

4. Segment Definition

Each geometrical segment input record contains:

a. A segment number.
b. A material number indicating the material contained in the segment.
c. A volume.
d. As many boundary numbers as required to uniquely define the segment.

a. Segment Number

Each segment is defined by a positive integer greater than zero and less than the maximum number of segments allowed (e.g. 1, 25, 47).

b. Material Number

The material number refers to the specific input record defining that material. See Material Definition.

c. Volume

The volume in cm^3 should be supplied for each segment. When the volume is not supplied, the code will attempt to calculate the volume of the segment. However, this calculation is possible only for certain simple shapes. The volume is used to display in the output the energy absorbed per unit volume.

d. Boundaries

A list of all required boundaries of the segment must be supplied as boundary numbers. Each boundary number refers to a Boundary Definition input record. See Boundary Definition. The overlay structure of the geometry description drastically reduces the number of boundaries that must be specified for a segment. Specification of unnecessary boundaries will slow down the calculation. When a boundary number is positive (i.e. no sign), the segment lies inside of a curved boundary or on the origin side of a plane. When a boundary number is negative (i.e. preceded by a minus sign), the segment lies outside of a curved boundary or on the side of the plane opposite of the origin. The sequence of the boundary numbers is unimportant.

5. Boundary Definition

Each boundary definition input record contains:

a. A boundary number
b. A boundary type
c. Geometrical data
d. A reflection type number

a. Boundary Number

Each boundary is defined by a positive integer greater than zero and less than the maximum number of boundaries allowed (e.g. 1, 25, 57). Several segments may refer to the same boundary number.

b. Boundary Type

Each boundary is defined by a 6 character field that describes the data format of the geometrical data (e.g. PLANE, CONIC, CYLIN). This feature allows great flexibility in data formats, e.g. one could specify that PLANEX defined a plane at right angles with the x-coordinate; only one geometrical datum is required in that case.

c. Geometrical Data

The specific geometrical data on a boundary input record are defined by the boundary type. In general they consist of direction cosines, a distance, a radius or a point. (e.g. for a PLANE they consist of the direction cosines of the normal of the plane away from the origin and the distance from the plane to the origin.)

d. Reflection Type Number

Each reflection type is defined by a positive integer greater than zero and less than the maximum number of reflection types allowed (e.g. 1, 5). Each reflection type may be defined by its own input record, but certain standard types are built into the code and are always available. (e.g. total reflection at boundaries of symmetry, Fresnel's laws of reflection from dielectrics, no reflections at all at dummy boundaries.)

6. Material Definition

Each material definition input record contains:

a. A material number.
b. An absorption coefficient.
c. An index of refraction.

a. Material Number

Each material is defined by a positive integer greater than zero and less than the maximum number of materials allowed (e.g. 1, 10). Several segments may refer to the same material number.

b. Absorption Coefficient

Each material must have an absorption coefficient α of units cm^{-1}. This coefficient will be used to calculate the intensity I of a ray of initial intensity I_0 after passing through x cm of the current material: $I=I_0e^{-\alpha x}$.

c. Index of Refraction

Each material must have an index of refraction that is larger than 1.0. When the indices of refraction of two adjacent materials differ by less than 10^{-4}, there will be no reflection at the boundary between those two materials. This rule reduces the number of boundary definitions that are necessary for boundaries that are partially real and partially dummy boundaries.

7. Reflection Definition

Each reflection definition record contains:

a. A reflection type number
b. A reflection data type
c. Reflection data

a. Reflection Type Number

The reflection of each boundary is defined by a positive integer greater than zero and less than the maximum number of reflection types allowed (e.g. 11, 25). Several boundaries may refer to the same reflection type number. The reflection type numbers between zero and ten are reserved for standard, built-in reflection types. The code currently contains the following reflection types:

a. Total reflection at boundaries of symmetry.
b. Periodic reflection for infinite dimensions.
c. No reflection for dummy boundaries.
d. Fresnel's laws of reflection for dielectrics.

b. Reflection Data Type

Each set of reflection data is defined by its data type that describes the data format of the reflection data. Currently only one data type has been implemented: constant reflection, independent of angle of incidence.

c. Reflection Data

The specific reflection data are defined by the reflection data type. They might consist of arguments to a function or coefficients to an equation. Currently only a constant reflection is allowed.